

## Maciej TROJNACKI

POLITECHNIKA RZESZOWSKA, WYDZIAŁ BUDOWY MASZYN I LOTNICTWA, KATEDRA MECHANIKI STOSOWANEJ I ROBOTYKI

# Szybkie prototypowanie adaptacyjnego algorytmu sterowania mini-robotem.

Dr inż. Maciej TROJNACKI

Urodził się w 1974 r. w Mielcu. Ukończył studia na Wydziale Budowy Maszyn i Lotnictwa Politechniki Rzeszowskiej w 1999 r. Pracę doktorską obronił również na PRz w 2003 r. Obecnie pracuje na stanowisku adiunkta w Katedrze Mechaniki Stosowanej i Robotyki PRz. Główne zainteresowania dotyczą sterowania ruchem naddążnym mobilnych robotów z zastosowaniem sieci neuronowych. Jest autorem kilku publikacji krajowych i międzynarodowych.



### Streszczenie

W pracy zaprezentowano środowisko szybkiego prototypowania adaptacyjnego algorytmu sterowania minirobotem. Środowisko to oparte jest ono na pakiecie Matlab/Simulink i przyborniku Real Time Windows Target. W referacie opisano sposób komunikacji z robotem, korzystający z portu LPT komputera PC. W badaniach doświadczalnych wykorzystano adaptacyjny algorytm sterowania, którego syntezę przeprowadzono na podstawie teorii stabilności Lapunowa.

### Abstract

In this paper a rapid prototyping environment of adaptive control algorithm for minirobot is presented. This environment is based on Matlab/Simulink package and Real Time Windows Target toolbox. In the work the communication way with mini-robot using personal computer LPT port is described. In experimental tests the adaptive control algorithm was used. Its synthesis was realized on the basis Lapunov stability theory.

## 1. Wstęp

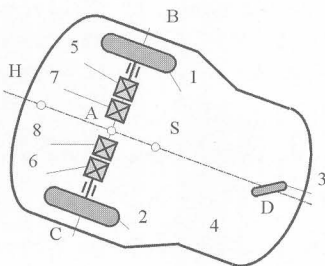
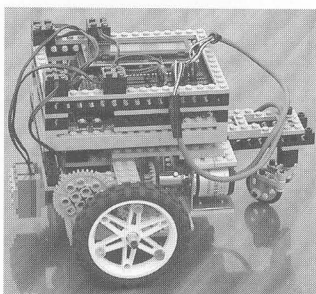
Środowiska szybkiego prototypowania oparte na specjalizowanych kartach np. firmy dSPACE [1, 2, 4] wiążą się zazwyczaj z dużymi kosztami, dlatego dostęp do tego typu środowisk przez większe grupy studentów jest ograniczony. Również z dużymi kosztami wiąże się zakup profesjonalnych robotów.

W pracy omówiono prototypową konstrukcję minirobotu 2-kołowego opartą na kontrolerze Handy Board oraz koncepcję środowiska, które do komunikacji robota z komputerem wykorzystuje porty LPT komputera PC. Omawiane środowisko nie wymaga dodatkowych urządzeń, poza ewentualnymi dodatkowymi kartami LPT. Dzięki temu jest możliwe szybkie i tanie wykonanie środowiska szybkiego prototypowania.

## 2. Minirobot 2-kołowy

Minirobot 2-kołowy pokazany na rys. 2.1a został skonstruowany w Katedrze Mechaniki Stosowanej i Robotyki Politechniki Rzeszowskiej z wykorzystaniem:

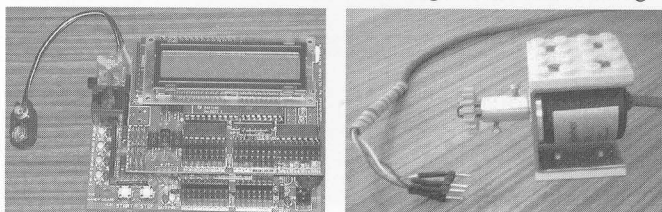
- zestawu Robotics Invention System 2.0,
  - kontrolera Handy Board z rozszerzeniem Expansion Board,
  - 2 enkoderów optycznych serii MOZ o rozdzielczości 100 dz/obr.
- Model takiego robota pokazano na rys. 2.1b.



Rys. 2.1. Minirobot 2-kołowy (a) oraz jego model (b).

Widoczne na rys. 2.1b oznaczenia mają następujące znaczenie:  
 A, B, C, D, H, S - charakterystyczne punkty robota [1, 3, 7, 8],  
 1, 2 - koła napędzające,  
 3 - samonastawne koło podpierające,  
 4 - rama robota,  
 5, 6 - silniki prądu stałego (9V),  
 7, 8 - enkodery optyczne MOZ 30/100/5.

Kontroler Handy Board wraz z rozszerzeniem Expansion Board pokazany na rys. 2.2a oparty jest na procesorze Motorola 68HC11, posiada szereg wejść analogowych i cyfrowych, wyjść cyfrowych oraz wyjść służących do sterowania silnikami prądu stałego. Umożliwia on również transmisję z komputerem PC za pomocą portu szeregowego. Połączenie pomiędzy kontrolerem, a komputerem PC odbywa się za pośrednictwem standardowego kabla telefonicznego.



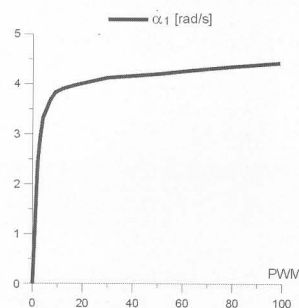
Rys. 2.2. Kontroler Handy Board (a), enkoder optyczny MOZ 30/100/5 (b)

Kontroler zasilany jest zestawem 8 akumulatorów AA. Dzięki temu, że kontroler posiada odpowiednią ilość wejść i wyjść cyfrowych możliwa jest również komunikacja równoległa z komputerem PC za pośrednictwem złącza LPT.

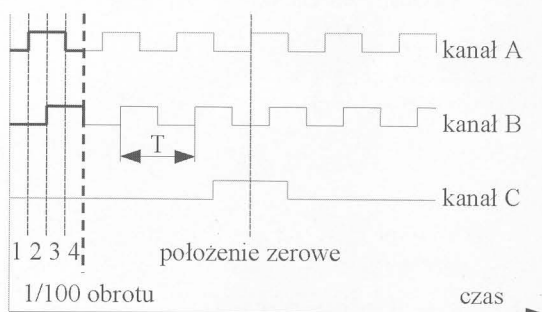
Oprogramowanie kontrolera Handy Board umożliwia zaprogramowanie jego w języku C i/lub w assemblerze.

Przed zaprojektowaniem środowiska szybkiego prototypowania przeprowadzono badania polegające na zbadaniu zespołu napędowego robota, wyznaczono m.in. charakterystykę prędkości obrotowej koła w funkcji sterowania PWM. W wyniku badań otrzymano charakterystykę pokazaną na rys. 2.3a.

Z charakterystyki tej wynika, że korzystne jest, aby w trakcie realizacji ruchu badany silnik pracował w zakresie 0-20 PWM, gdyż powyżej tego zakresu zmiana prędkości obrotowej silnika przy dużej zmianie sterowania jest niewielka.



Rys. 2.3.a Charakterystyka prędkości kątowej koła minirobotu w funkcji sterowania PWM



Rys. 2.3.b Zasada działania enkodera MOZ 30/100/5.

Przeprowadzono również badania polegające na porównaniu dokładności różnych algorytmów pomiaru prędkości obrotowej za pomocą enkoderów optycznych. Dokonując pomiaru prędkości obrotowej enkodera wykorzystuje się przerwania procesora generowane z częstotliwością 1kHz.

Zastosowane w minirobocie enkodery MOZ 30/100/5 posiadają 3 kanały oznaczone jako A, B i C (rys. 2.3b). Kąt obrotu enkodera można określić z dokładnością do 1/400 obrotu, czyli 0.9o, natomiast dzięki temu, że kanały A i B są przesunięte względem siebie o 1/4 okresu można również określić kierunek obrotu enkodera.

Należy również zwrócić uwagę na dokładność pomiaru kąta obrotu kół z zastosowaniem enkoderów. W przypadku minirobota 2-kołowego przy zastosowaniu enkoderów MOZ 30/100/5 dokładność ta wynosi:

$$\epsilon = 2\pi / 400 \cdot 3/5 \approx 0,01 \text{ [rad]}$$

Do opisu dynamiki minirobota 2-kołowego (rys. 2.1a) wykorzystuje się równania Maggieo, które w zapisie macierzowo-wektorowym mają postać [1, 2]:

$$M(q, \dot{q})\ddot{q} + C(q, \dot{q}, a)\dot{q} + F(\dot{q}, a) = u \quad (2.1)$$

### 3. Adaptacyjny algorytm sterowania minirobotem

W niniejszym punkcie przeprowadzono syntezę adaptacyjnego algorytmu sterowania ruchem nadążnym minirobota 2-kołowego.

Przez pojęcie ruchu nadążnego rozumie się ruch wybranego punktu mobilnego robota kołowego po zadanej trajektorii ruchu.

W syntezie adaptacyjnego układu sterowania ruchem nadążnym minirobota 2-kołowego zakłada się, że wybrany punkt H minirobota przemieszcza się po trajektorii zadanej  $q_d(t)$ , takiej, że sygnały  $q_d(t)$ ,  $\dot{q}_d(t)$  i  $\ddot{q}_d(t)$  są sygnałami ograniczonymi oraz spełnione jest równanie więzów nieholonomicznych  $J(q_d)\dot{q}_d = 0$ , a ponadto układ jest obserwowalny.

Należy wyznaczyć prawo sterowania u i prawo adaptacji wektora parametrów  $\hat{a}$ , takie, że dla warunków początkowych  $[q(0), \dot{q}(0)] \in \Omega = \{[q, \dot{q}] : J(q)\dot{q} = 0\}$  rozwiązanie  $q[t, q(0), \dot{q}(0)]$ ,  $\dot{q}[t, q(0), \dot{q}(0)]$  jest zbieżne do żądaności:

$$\Omega = \{[q, \dot{q}] : q = q_d, \dot{q} = \dot{q}_d\} \quad (3.1)$$

Wektor błędu nadążania zdefiniowano jako:

$$q_e(t) = q(t) - q_d(t) \quad (3.2)$$

Syntezę adaptacyjnego układu sterowania przeprowadza się w oparciu o teorię stabilności Lapunowa przyjmując funkcję Lapunowa w postaci: [1,4]

$$V(t) = 1/2 [s^T M(q, \dot{q}) s + \tilde{a}^T \Gamma \tilde{a}] \quad (3.3)$$

gdzie:

$$\tilde{a} = \hat{a}(t) - a, \quad s = \dot{q} - v, \quad v = \dot{q}_d - \Lambda q_e$$

$\hat{a}$  - wektor ocen parametrów,  $\tilde{a}$  - błąd estymacji parametrów.

W wyniku przeprowadzonej syntezy przyjęto sygnał sterowania w postaci:

$$u = Y(v, \dot{v})\hat{a} - K_D s \quad (3.4)$$

Natomiast prawo adaptacji parametrów przyjęto jako:

$$\dot{\hat{a}} = -\Gamma^{-1} Y^T(v, \dot{v}) s \quad (3.5)$$

Ostatecznie otrzymano zależność:

$$\dot{V}(t) = -s^T K_D s \leq 0 \quad (3.6)$$

z której wynika, że funkcja V jest funkcją Lapunowa, a tak przyjęte prawo sterowania (3.4) zapewnia globalną stabilność w sensie Lapunowa.

### 4. Środowisko szybkiego prototypowania dla minirobota

Dla minirobota 2-kołowego zaprojektowano stanowisko do badań eksperymentalnych oparte na środowisku Matlab/Simulink oraz wykorzystujące dodatkowo przybory Real Time Workshop, Real Time Windows Target [5] i kompilator Visual C++ 6.0. Śro-

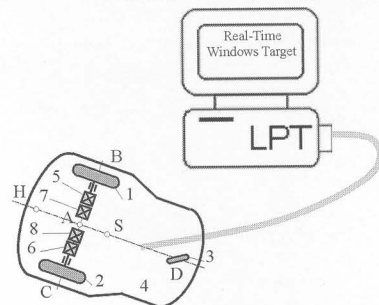
dowisko to zostało zastosowane do szybkiego prototypowania adaptacyjnego algorytmu sterowania mini-robotem 2-kołowym.

Do komunikacji komputera ze środowiskiem Real Time Windows Target, na którym zaimplementowany jest algorytm sterowania, a obiektem rzeczywistym korzysta się ze złącza LPT. Schemat takiego środowiska został pokazany na rys. 4.1a.

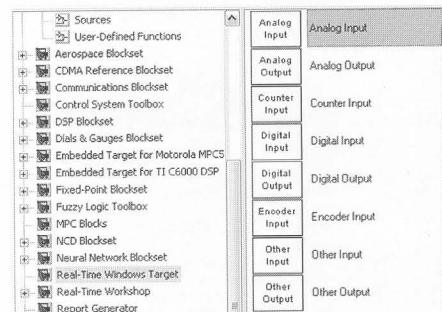
Przybory Real Time Windows Target [5] umożliwia realizację algorytmu sterowania w czasie rzeczywistym w środowisku Windows.

Zawiera on bloki (rys. 4.1b) umożliwiające korzystanie z urządzeń wejścia-wyjścia takie jak: wejścia i wyjścia analogowe, wejścia i wyjścia cyfrowe, liczniki, wejścia dla enkoderów, inne urządzenia wejścia/wyjścia np. specjalizowane karty.

Przykładowo korzystając z tego pakietu można w układzie sterowania wykorzystać takie elementy: joystick, myszkę, porty LPT i enkodery.



Rys. 4.1.a Środowisko szybkiego prototypowania dla minirobota oparte na przybory Real Time Windows Target

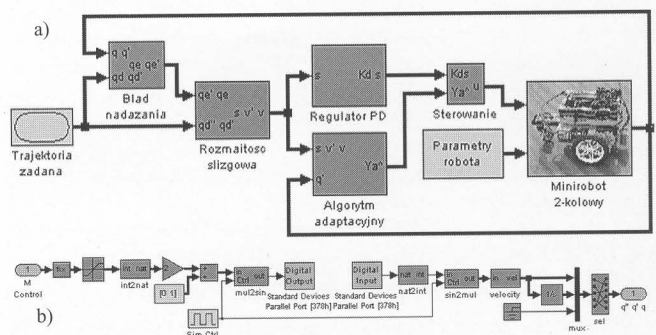


Rys. 4.1.b Środowisko szybkiego prototypowania dla minirobota przybory Real Time Windows Target

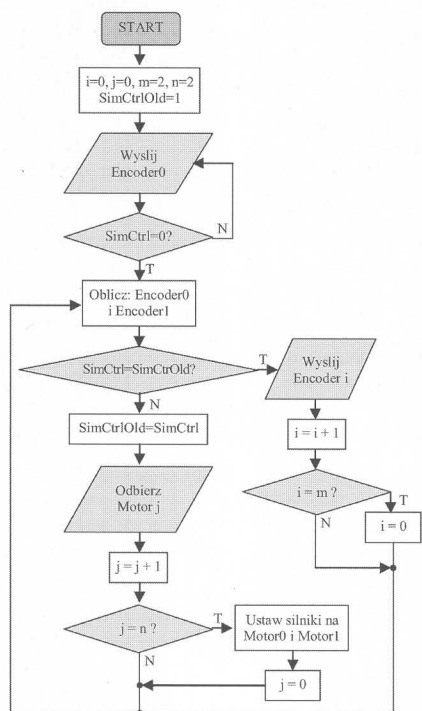
Prezentowane środowisko szybkiego prototypowania składa się z dwóch podstawowych elementów:

- komputera PC, na którym w środowisku Matlab/Simulink/Real Time Windows Target zaimplementowany jest układ sterowania robotem,
- robota zawierającego program w assemblerze, służący do komunikacji z komputerem PC, przesyłania informacji z enkoderów i ustawiania wyjść PWM do sterowania silnikami na podstawie sygnałów z komputera PC.

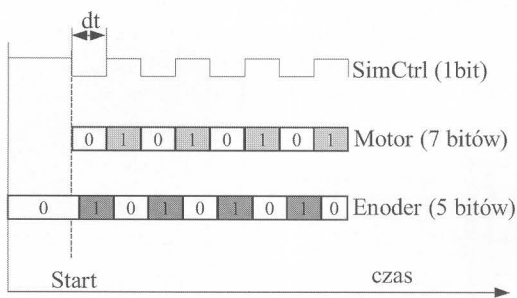
Schemat układu sterowania zaimplementowanego na komputerze PC pokazano na rys. 4.2a. Natomiast na rys. 4.2b pokazano sposób realizacji komunikacji pomiędzy komputerem a minirobotem. Przyjęto krok dyskretyzacji  $dt = 4 \text{ ms}$ .



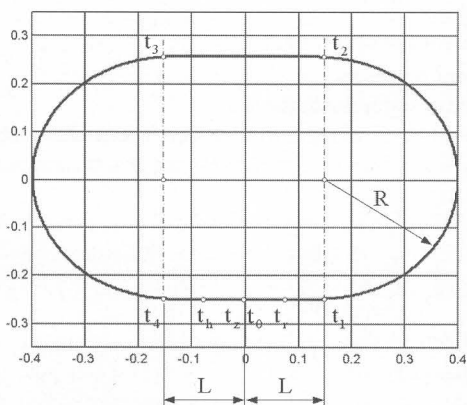
Rys. 4.2. Schemat układu sterowania robotem (a), realizacja komunikacji z robotem (b)



Rys. 4.3.a Algorytm realizowany przez kontroler HB



Rys. 4.3.b graficzna ilustracja komunikacji kontrolera z Simulinkiem



Rys. 4.3.c założony tor ruchu punktu H minirobota

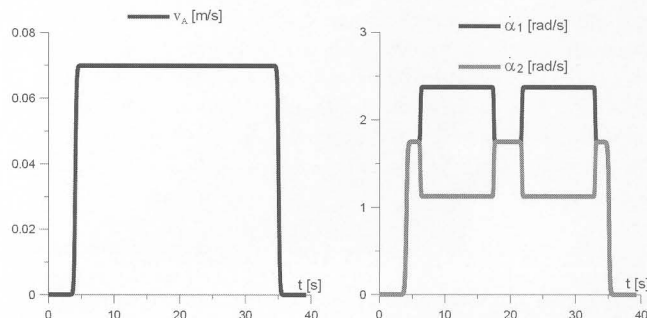
Natomiast na rys. 4.3a pokazano schemat algorytmu realizowanego przez kontroler Handy Board. Przed uruchomieniem programu inicjalizowane jest wyjście portu LPT, na którym najmniej znaczący bit ustawiany jest w stan wysoki (zmienna SimCtrl = 1), następnie uruchamiany jest program zaimplementowany na kontrolerze HB. Program ten został napisany w assemblerze, główna pętla programu realizowana jest w przerwaniach procesora co 1 ms (po SimCtrl = 0).

### 5. Szybkie prototypowanie neuronowego algorytmu sterowania minirobotem

W oparciu o opis dynamiki minirobota 2-kołowego (2.1) i syntezę adaptacyjnego układu sterowania ruchem nadążnym dla tego robota, wykonano badania doświadczalne wykorzystu-

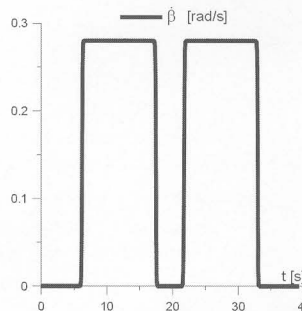
jące środowisko szybkiego prototypowania omówione w poprzednim punkcie referatu.

Jako trajektorię do sterowania adaptacyjnego ruchem nadążnym minirobotem 2-kołowego przyjęto trajektorię pokazaną na rys. 4.3c, natomiast na rys. 5.1 pokazano zadane przebiegi parametrów ruchu robota.



Rys. 5.1.a Zadane przebiegi parametrów ruchu minirobota: prędkość charakterystycznego punktu A

Rys. 5.1.b Zadane przebiegi parametrów ruchu minirobota: prędkości kątowe obrotu własnego kół napędzających



Rys. 5.1.c Zadane przebiegi parametrów ruchu minirobota: prędkość kątowa obrotu własnego ramy

W celu ilościowej oceny algorytmu sterowania, wprowadzono następujące wskaźniki jakości:

- całka kwadratu błędu nadążania:

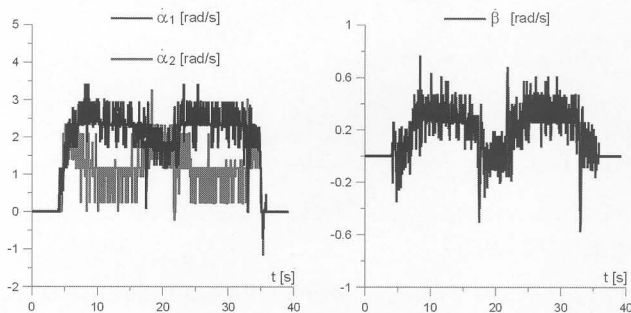
$$E = \int_0^T q_e^2 dt \quad (5.1)$$

- maksymalna bezwzględna wartość błędu nadążania:

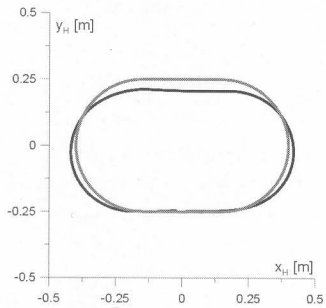
$$q_{e\max} = \max_{t \in E} (|q_e|) \quad (5.2)$$

W wyniku szybkiego prototypowania adaptacyjnego algorytmu sterowania minirobotem otrzymano przebiegi pokazane na rys. 5.2 i 5.3.

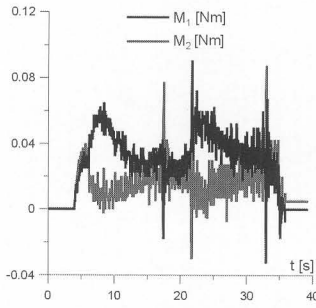
Dla porównania błędów uzyskanych dla układu adaptacyjnego przeprowadzono weryfikację bez sterowania kompensacyjnego. Uzyskane przebiegi pokazano na rys. 5.4. Z porównania obu przebiegów i uzyskanych wskaźników jakości wynika, że zastosowane sterowanie adaptacyjne zapewnia większą dokładność sterowania ruchem nadążnym minirobotem 2-kołowego.



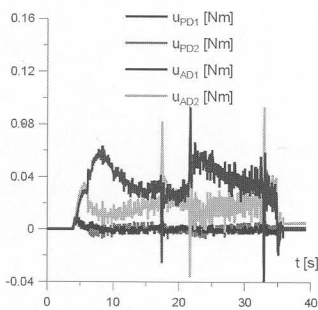
Rys. 5.2a Przebiegi parametrów ruchu minirobotem 2-kołowego: prędkości kątowe własnego ramy  
Rys. 5.2b prędkość kątowa obrotu własnego kół napędzających



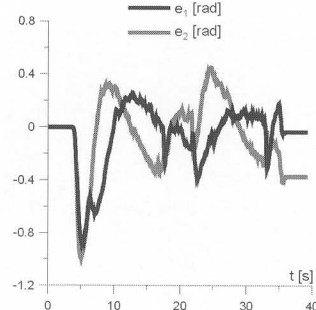
Rys. 5.2c tor ruchu charakterystycznego punktu H (linia czerwona - zadany, czarna - otrzymany w wyniku weryfikacji).



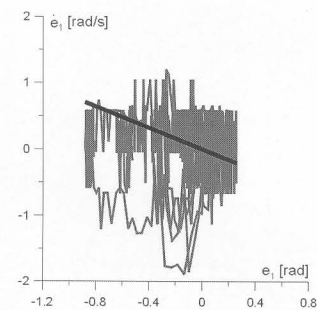
Rys. 5.3a Momenty napędzające



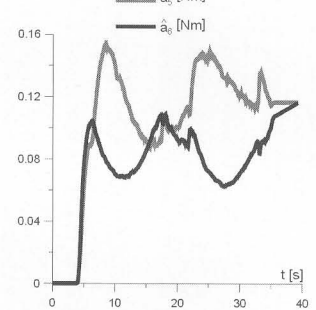
Rys. 5.3b sterowanie kompensacyjne i sterowanie PD



Rys. 5.3c błędy nadążania



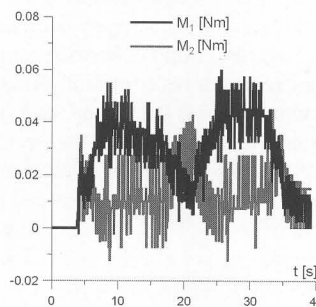
Rys. 5.3d rozmaitość ślizgowa



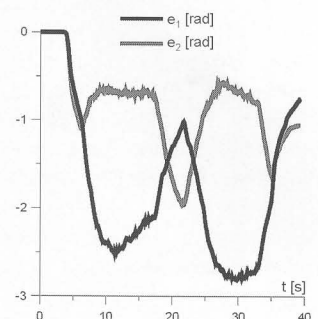
Rys. 5.3e wybrane przebiegi ocen parametrów robota

nr koła	E	q <sub>emax</sub>
1	2.3917	0.8841
2	3.3201	0.9981
<b>w. śred.</b>	<b>2.8559</b>	<b>0.9411</b>

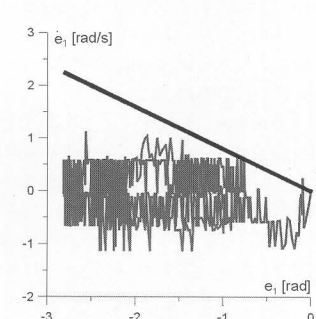
nr koła	E	q <sub>emax</sub>
1	146.37	2.8141
2	37.82	1.9712
<b>w. śred.</b>	<b>92.10</b>	<b>2.3927</b>



Rys. 5.4a Momenty napędzające dla regulatora adaptacyjnego (f) i regulatora PD (g)



Rys. 5.4bc Momenty napędzające błędy nadążania (b,c) dla sterowania z regulatorem PD



## 6. Uwagi końcowe

Z uzyskanych wyników badań metodą szybkiego prototypowania wynika, że dokładność sterowania uzyskana z zastosowania regulatora PD, jest mniejsza w stosunku do rozwiązania, w którym stosuje się dodatkowo sterowanie adaptacyjne, czyli uwzględnia się kompensację nieliniowości badanego obiektu i zakłócenia parametryczne.

Największe błędy w realizacji założonego ruchu występują w okresie rozruchu i hamowania. Błędy w trakcie rozruchu wynikają z faktu, że w tym okresie nie występuje kompensacja własności dynamicznych mobilnego robota, gdyż dopiero rozpoczyna się proces adaptacji parametrów obiektu. Natomiast występowanie błędów nadążania parametrów ruchu w okresie hamowania, wynika ze zmiany trajektorii zadanej. Błąd ten jest szybko kompensowany przez algorytm sterowania. Ocena układu sterowania w analizowanych okresach ruchu, potwierdza również przebieg trajektorii fazowej. Z otrzymanego ruchu układu wokół rozmaitości ślizgowej wynika, że testowany adaptacyjny algorytm sterowania jest odporny na występujące zakłócenia.

Zaprezentowane w niniejszym referacie środowisko szybkiego prototypowania oparte na pakiecie Matlab/Simulink/Real Time Windows Target umożliwia szybkie i tanie wykonanie badań doświadczalnych dla prototypowych konstrukcji robotów. Omawiane środowisko można rozbudować zwiększając ilość przesyłanych danych pomiędzy robotem a komputerem poprzez dodanie dodatkowych kart z portami LPT lub rozszerzeniu ramki danych i wydłużeniu czasu jej przesyłania. Możliwe jest również zmniejszenie czasu dyskretyzacji np. do 2 ms.

## Literatura

- Giergiel J., Hendzel Z., Żylski W.: Kinematyka, dynamika i sterowanie mobilnych robotów kołowych w ujęciu mechatronicznym, Monografie, Wyd. IMiR, AGH, Kraków, 2000.
- Giergiel M. J., Hendzel Z., Żylski W.: Modelowanie i sterowanie mobilnych robotów kołowych, PWN, Warszawa, 2002.
- Hendzel Z.: Sterowanie ruchem nadążnym mobilnych robotów kołowych, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów, 1996.
- Hendzel Z., Trojnecki M.: Szybkie prototypowanie adaptacyjnego algorytmu sterowania mobilnym robotem, Projektowanie Mechatroniczne, Praca zbior. pod red. T. Uhla, Wydawnictwo Katedry Robotyki i Dynamiki Maszyn, AGH, Kraków, 2002, str. 90-99.
- Real-Time Windows Target® User's Guide, Version 2, The MathWorks, Inc., 1999-2000.
- Spong M.W., Vidyasagar M.: Dynamika i sterowanie robotów, WNT, Warszawa, 1997.
- Trojnecki M.: Sterowanie ruchem nadążnym mobilnego robota kołowego z zastosowaniem sieci neuronowych, Rozprawa doktorska, PRZ, Rzeszów 2003.
- Żylski W.: Kinematyka i dynamika mobilnych robotów kołowych, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów, 1996.

Praca wykonana w ramach realizacji grantu projektu badawczego nr 5T07C00722.

**Title:** Fast prototyping of adaptive control of a mobile mini robot.